Computer Networks 174 (2020) 107222

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

NC based DAS-NDN: Network Coding for robust Dynamic Adaptive Streaming over NDN

Muhammad Hassan[a], Mauro Conti[a], Chhagan Lal[b,*]

[a] Department of Mathematics, University of Padua, Padua, Italy
[b] Simula Research Laboratory, Norway

ARTICLE INFO

Keywords:
Name Data Networking
Adaptive multimedia streaming
Bit rate oscillations
DASH
Network Coding

ABSTRACT

Over the last few years, the dramatic growth in video demand has inspired the service providers (e.g., Netflix and YouTube) to swing towards HTTP based Dynamic Adaptive Streaming (DASH). However, sustaining the adequate bandwidth claims over this rapid growth in multimedia content becomes a significant challenge for network operators. Considering the effectiveness of the next generation future Internet architecture, i.e., Name Data Networking (NDN), recently DASH over NDN is implemented. The fundamental characteristics of NDN, such as efficient content distribution and low bandwidth requirements, significantly increase the bandwidth utilization, which ensures the smooth delivery of multimedia content. However, we discovered that the above characteristics of NDN also opens the door for new vulnerabilities.

In this paper, first we propose a new attack termed as "Bitrate Oscillation Attack" (BOA), which disrupt the functionality of DASH protocol over NDN by exploiting its two key features called *in-network caching* and *interest aggregation*. In particular, BOA forces the DASH streaming system running at the honest client to oscillate in various video resolutions with high frequency and amplitude, within a single video session. Second, to mitigate the BOA, we design and implement a proactive countermeasure called "NC based DAS-NDN". Our solution efficiently enables the network coding to DAS multimedia content and within NDN architecture. Thus, without any coordination between the network nodes reduces bitrate oscillations in the presence of BOA and NDN's inherent content source variations. The performance evaluation performed on different target scenarios proves the effectiveness of our proposed attack, and the results also show the correctness of our proposed corresponding countermeasure. In particular, the result analysis shows that BOA increases the annoyance factor in spatial dimension of end-user, and our countermeasure greatly reduces the adverse effects of BOA and also make DAS friendly to NDN's inherent features.

1. Introduction

The significance and usage of multimedia traffic are rapidly increasing from the last few years. It is due to the technological advancements in mobile devices, hardware, and design & deployment of new networking technologies (e.g., LTE, 5G, software-defined networks, and cloud computing) which supports the efficient multimedia usage. As per the Cisco Visual Networking Index (VNI) predictions [1], the video data will occupy approximately 82% of the IP traffic, and the mobile devices alone will contribute about one-third of Internet traffic by 2022. At present, multimedia streaming providers like YouTube and Netflix together produce approximately 50% of IP traffic. The other similar service providers like Hulu, Amazon-Prime, and HBO-GO are also gaining rapid popularity. This exponential growth in the multimedia traffic over the current Internet architecture causes various challenges for network operators mainly about satisfying the user requirements (e.g., bandwidth, and latency). Taking into account the rapid growth in internet traffic, the researchers are proposing new networking paradigms (e.g., Name Data Networking (NDN) [2–5]) to meet the bandwidth requirements of end-users. By caching the data within the network, NDN handles the congestion, scales the Internet, and ease the bandwidth blockage. In NDN, the routers not only perform the routing operations but they also perform in-network caching and interest aggregation [3,6]. In particular, NDN provides built-in support to cache and multi-cast the data at network layer devices. Unlike IP, in which routers mainly do the routing of data packets, NDN routers forwards and cache the incoming data packets. In this way, NDN architecture replaces the communication model from host-centric to content-centric, and it is widely accepted as a Future Internet Architecture (FIA) [7].

Currently, the content providers uses HTTP based Dynamic Adaptive Streaming (DASH) [8–10] as a primary protocol for multimedia

* Corresponding author.
E-mail addresses: hassan@math.unipd.it (M. Hassan), conti@math.unipd.it (M. Conti), chhagan@simula.no (C. Lal).

https://doi.org/10.1016/j.comnet.2020.107222
Received 26 September 2019; Received in revised form 10 January 2020; Accepted 15 March 2020
Available online 22 March 2020

traffic delivery due to its optimize network bandwidth usage feature. In DASH, to attain best possible Quality of Experience (QoE) during a video streaming session, the user dynamically adapts to the most appropriate resolution (i.e., bitrate) based on certain network and system properties such as link bandwidth, latency, and system buffer size.

Hence, DASH is becoming a standard protocol for streaming multimedia traffic over the Internet. In DASH, the adaptation logic makes decisions based on the real-time measurements of link bandwidth or latency. In this way, DASH moves the accountability of quality assurance from network or producer to the consumer side in a transparent way [11]. By taking into account, the effectiveness of NDN paradigm which minimizes the challenges faced by the existing Internet, and to overcome limitations of multimedia streaming, recently, research community investigated the implementation of Dynamic Adaptive Streaming (DAS) over NDN [12–15]. Several studies, e.g., [16–20], evidently highlights the efficacy of this combination, where NDN's inherent characteristics provide support to DAS, thus, makes DAS a perfect candidate for efficient and scalable multimedia streaming over NDN. NDN's in-network caching and multipath transmission provides several benefits to the content provider, such as reduced bandwidth requirement and network latency. However, the direct mapping of DAS over NDN opens new security vulnerabilities that an attacker could exploit to reduce the QoE perceived by an end-user.

1.1. Motivation and Contribution

As NDN is still in its initial stages of recognition as a potential FIA, it becomes important to identify and address the potential security threats that might exist or arise due to the use of existing technologies (such DAS) in the NDN environment. To this end, we identified that during the adaptive bitrate streaming, an adversary could adversely exploit two fundamental characteristics of NDN, called *in-network caching* and *interest aggregation* [3]. More specifically, an attacker can target the bitrate adaptation logic unit of DAS that results in the degradation of QoE perceived by the benign user. To launch the attack, it is assumed that same as the benign users, the attacker also has access to the multimedia data through NDN routers, and he/she knows in advance about the multimedia content that the benign consumer will request in the near future. By implementing our proposed algorithm, the attacker is able to launch a Bitrate Oscillation Attack (BOA) which forces the DAS client to compute false bandwidth estimations during bitrate adaption process, causing video streaming with highly variable bitrates. The primarily results verifying the adverse effects of our proposed BOA attack on user perceived QoE is reported in [21]. As per our knowledge, this is the first such attack on NDN along with its countermeasure. We believe that it is essential to carefully investigate the DAS functionality (and change it as needed) before its actual deployment in NDN for real-world multimedia services.

Although NDN naively supports multipath routing and content distribution without the need of any supplementary protocol [3]. For example, the interests issued by the client to retrieve data packets can be replied by any network node which cache the content. However, NDN does not provide any mechanism to provide coordination about the cached content and forwarding strategies between the network nodes. Due to the lack of any coordination scheme, the interest packets sent for the same content might result in retrieval of that content from multiple sources without considering the requirements of a particular application, e.g., DAS. Since the DAS client receiving the multimedia content from multiple paths results in sub-optimal bitrate adaptation (i.e., degrades perceived QoE). In contrast, providing coordination between forwarding strategies and cache states would require global knowledge of the complete network. It results in increased computational complexity and makes it hard to scale in large and dynamic networks.

Apart from extending the attack scenarios and result analysis that are provided in [21], this paper proposes a novel mitigation technique called as Network Coding enabled DAS over NDN (NC based DAS-NDN), which is a network-side solution to countermeasure the BOA. In particular, we propose a robust framework for adaptive bitrate streaming by efficiently enabling network coding [22–24] in NDN to provide optimal multipath multimedia content delivery for DAS clients. Our proposed scheme retains the functionality of ICN's inherent features, i.e., in-network caching and multipath transmission, but, it sends network coded interest packets to request network coded data packets. These data packets are produced from the same set of original data packets while considering the parameters of QoE perceived by the DAS clients. Thus, NC based DAS-NDN aims to reduce bitrate oscillations caused by an attacker or NDN's inherent content source variations without coordinating the forwarding strategies (and cache states) between the network nodes. In summary, the significant contributions of our work are the following.

- We propose a novel attack called *Bitrate Oscillation Attack* (BOA). During a video streaming session, the BOA makes the benign consumer(s) to perform the streaming in different resolutions/representations, thus decreases the user QoE. To perform the attack, the attacker exploits the adaptation logic of DAS along with the inherent characteristics of NDN routers. Through implementation and result analysis, we prove the effectiveness, (i.e., higher annoyance factor in the spatial dimension of user), of our proposed attack algorithm.
- We propose *NC based DAS-NDN*, an effective countermeasure to BOA. It is a network coding enabled technique for DAS over NDN to resist the bitrate oscillations caused by either the adversary or the inherent content source variations, with any coordination between the network nodes. We implement a new model of NDN network node which supports the functionalities of network coding. We also present a new design of network coding enabled client and content source, which efficiently adopts network coded multimedia content in order to enhance the performance of DAS.
- We fully implement our attack and its proposed countermeasure using the NDN network simulator called AMuSt-ndnSIM [25]. The performance evaluation done in different target NDN scenarios show the impact of the attack on user's perceived QoE, and how our countermeasure resists the attack to improve the perceived QoE.

1.2. Organization

The remaining of this paper is organized as follow. DASH protocol is discussed along with the state-of-the-art on the DASH over NDN in Section 2. The system and adversary models on which our attack and its countermeasure is evaluated is presented in Section 3. The description of our proposed attack is given in Section 4, and the working methodology of the proposed mitigation approach given in Section 6. The simulation setup and performance evaluation through result analysis is presented in Section 7. Finally, in Section 8, we conclude our work with future research directions.

2. Related Work

Recently, DAS has been developed as a most used technique for adaptive bitrate streaming mechanism which supports real-time or on-demand video streaming. Therefore, almost all the famous providers in the market of Internet video streaming such as Netflix, Amazon prime, and YouTube, relies on DAS [13]. All these service applications uses MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [9,10], which is endorsed by ISO/IEC and it turn out to be the highest used solution for DAS. In particular, DASH lay down the description of video data availability, and the procedure of how it should be segmented.

With the appreciation of NDN [3,6] as a possible future replacement of exiting Internet architecture, and the increasing popularity of DASH [26], the DASH over NDN implementation has gained noteworthy consideration from the research community. Numerous studies [13,14,16–19] have shown in-network functionalities presented by

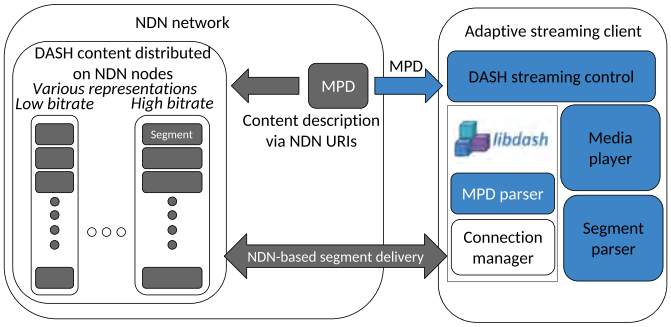

Fig. 1. Dynamic adaptive multimedia streaming over NDN

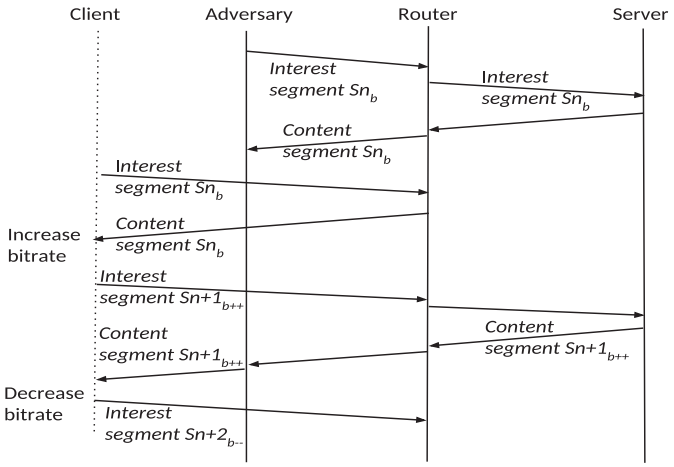

Fig. 2. Sequence of interests to launch BOA

NDN as a support for DAS. For example, authors in [20] demonstrate a combination of DASH and NDN by implementing a proxy service between HTTP and NDN, and authors in [16,19] completely exploit the potential of NDN and shows the implementation of DASH client as a instinctive NDN interface. In specific, the framework converts the HTTP request and reply messages to the corresponding NDN interest and content messages. Fig. 1 illustrate the proposed architecture of DASH over NDN [6], where DASH-related components are marked in light blue and NDN-related components in dark grey color. In NDN, similar to HTTP-based DAS, MPD defines the relationship between a segment's associated characteristics (i.e., bitrate, resolution, versions etc) and its name using NDN names, however, uses Uniform Resource Identifiers (URIs) instead of URLs [12]. In addition, the hierarchical naming scheme of NDN explicitly supports DASH versioning and segmentation [16]. The starring role of DASH's streaming control system is to adjust the client requests based on the available bitrates and estimated available network bandwidth. It is done to deliver a smooth streaming session to end users with high QoE.

The work in [16,17,19] confirms that DASH over NDN is able to provide enhanced performance in terms of average download bitrate, smooth streaming sessions, and reduced bandwidth requirements. Also, the outcome of [16,19] demonstrates the effectiveness of in-network caching in the case where multiple clients request similar contents, subsequently showing improved video quality over time. Furthermore, the authors in [18] exhibit the gain of NDN-based dynamic adaptive streaming while using Scalable Video Coding (SVC), proving that integration of the layered data approach with in-network caching increases the performance of bitrate adaptation process and provides smooth playback without stalling.

The QoE in adaptive video streaming is a fundamental factor which relies on the intermingling of high video quality (e.g., high bitrate) and high streaming performance (e.g., continuous playback without re-buffering). The authors in [27] and [28] explain the impairment factors that affect the user experience for dash video, and it illustrates that the frequent switching in video representations in a session diminishes streaming quality. Thus, the spatial quality of video can be determined by the level of variations occurred during a streaming session.

The role of DASH streaming control mechanism is to adapt the client requests based on the available bitrate and network bandwidth. In this way, it aims to provide a smooth streaming session with high Quality of Experience (QoE). In this paper, we observed the behaviour of DASH streaming control system while it interacts with NDN's implicit characteristics. Although the results from [13,14,16–20] show that NDN features are advantageous for adaptive video streaming, we show that the DASH streaming control system also exposes new security vulnerabilities when it intersects with NDN's architecture [21]. We claim that by exploiting these features a malicious user can degrade the QoE of a DASH client during its streaming process. In particular, we identify that an attacker can adversely exploit two ICN features, namely in-network caching and interest aggregation. In particular, the adversary can harm the adaptive behaviour of DASH streaming control system.

3. System and Adversary Models

In this work, we take a scenario of adaptive video streaming in NDN paradigm as it is shown in Fig. 2. The producer (P) publishes multimedia content which it stores in a DASH-compatible format. The multimedia file (say, S) consists of n number of segments of equal length, moreover, all the segments are available for streaming in various encoded bitrates (b), i.e., where each bitrate represents a different resolution/quality of the same content. To perform the video streaming, the client (C) requests (by sending appropriate interest messages) the segment(s) from producer in one of the available bitrate of S at P, the choice of requesting a particular bitrate depends upon the DASH adaptation logic unit. We assume that an adversary (Adv) have the advance knowledge about the content or video file that C will be streaming next or in near future. Lastly, the interest messages sent by both C and Adv travels via multiple NDN routers ($R_i \epsilon |R|$) to reach P Table 1 presents the summary of notations used in the paper.

3.1. System Model

In the evaluation scenario that we consider in this paper, the entities all including C, Adv, P, and R are configured with the NDN stack. To perform the adaptive streaming during a video fetch, C and P uses the traditional DASH over NDN protocol [16,19]. The Adv knows about the usage of DASH protocol in the network, and it chooses to attack (to degrade consumer QoE) the DASH functionality instead of using it to optimize its own streaming operations. The DASH-compliant multimedia streaming mainly uses the following two well known video encoding techniques, namely, Scalable Video Coding (SVC) [29], and Advanced Video Coding (AVC) [30]. In SVC technique, the video data is encoded in independent layers of quality known as base layer (BL) and enhancement layers (EL), in such a way that each layer subsequently contributes in the enhancement of video quality. While in AVC, all the segments which has a different bitrates are represented independently by a unique segment name. For example, a segment of bitrate 100 kb/s is represented as `/dash/bunny/_2s_100kbit/bunny_2s1.m4s`. The naming information of all the different bitrate segments is given in the MPD file through URI [12]. The consumer requests a segment in an appropriate bitrate that is calculated by the DASH's adaptive logic unit. We perform a thorough investigation of the exploitation that attacker performs in the network and its effects on the behaviour of C by using all forms of adaptation strategies that are given in traditional DASH streaming control system (i.e., Rate-Based (RB), Buffer-Based (BB), and Rate-Buffer-based

(*R&B*) [25]. Below, we briefly discuss the functionality these adaptation strategies.

3.1.1. Rate-Based adaptation logic

The *RB* adaptation algorithms [31,32] base on the idea of using the previous segment's measured bandwidth as a measure of bandwidth estimation of subsequent segment. In this way it measures the bandwidth availability of network links during streaming. Specifically, in RB technique, C calculates the link bandwidth at every instance while downloading a segment, and then it uses an exponential weighted moving average to calculate available bandwidth for the next segment, as illustrated in Eq. 1.

$$\lambda_{k+1} = (1 - \beta) * \lambda_k + \beta * \lambda, \tag{1}$$

where λ_{k+1} is the next bandwidth estimation and λ_k is the previous one. λ is calculated as a ratio of current segment size to its download time. β is a constant, which reduces the impact of fresh measures on the estimate. Using the Eq. 1, C select the highest affordable media encoded bitrate b_{k+1} for the next segment (i.e., $b_{k+1} < \lambda_{k+1}$) that it fetches.

3.1.2. Buffer based adaptation logic

The *BB* adaptation technique uses current buffer occupancy level, $B(t)$, at the client machine as a measure to estimate bitrate for next segment. The buffer is divided into multiple levels, and C requests b_{k+1} which depends on its actual buffer level. We use *Bandwidth independent Efficient Buffering* (BiEB) [33] scheme as a standard, and it has a maximum buffer limit of 33 seconds.

3.1.3. Rate and Buffer-Based Adaptation logic

The R&B adaptation technique [34] uses a hybrid approach by combining the *RB* and *BB* mechanisms. In R&B, C performs the bandwidth estimation for the next segment λ_{k+1} such that it stabilizes the buffer level $B(t)$ around a target value (B_{max}). The aim is to keep the adaptation rate smooth by avoiding its reaction to fluctuating link bandwidth. The R&B adaptation technique use two threshold values (B_{min} and B_{max}) along with λ_k and $\lambda_k + 1$. Then, the increase/decrease in bitrate selection process is governed in following two ways: (i) decrease, when $B(t) > B_{max}$, and when $B_{min} \leq B(t) \leq B_{max}$, quickly shift to a lower quality, and finally, when $B(t) < B_{min}$, request the lowest quality, and (ii) increase, if $B_{min} \leq B(t) \leq B_{max}$ or greater than B_{max}.

3.2. Adversary model

For the adversarial model, we first assume that Adv is connected to at least one on-path router between C and P. It is not a strict assumption as the adversary could be connected to any intermediate router(s) that are part of the route between P and C. Using the state-of-the-art geo-locating techniques such as [35], an Adv can identify the routers that are close to C and then connect with them. In our performance evaluation section, we show that how the different positions of Adv with respect to C effects the QoE of video streaming at C. In particular, in Section 7, we show the adverse impact on C when a Adv launches the attack by connecting to different locations in the network. Also, we show the impact of attack using different launching techniques by Adv.

Other assumption we made is that Adv has prior information about the multimedia data (S) which C will be streaming in close future. Many state-of-the-art schemes support our assumption apart from the preliminary knowledge required to execute the attack subjective to C. For instance, Adv can exploit the timing attacks as a side channels to breach the consumer privacy and infer if some content has been previously streamed by C [36]. Additionally, Adv could also specifically probe a MPD file by exploiting the timing attacks to discover whether C has previously requested a content (i.e., video related to MPD file) or not. These schemes allow Adv to predict the multimedia data that C is or will be requesting. Also, it could be the case that C and Adv share the same wireless connection, which allows the adversary to trace network traffic and it can easily perform the eavesdropping attack [37]. Based on the traffic traces of the consumer, Adv could have some knowledge about the online activities of C, and it could predict the streaming content and its source [38].

Finally, a viral video file is always a good candidate to launch our BOA in the target network. In this case, the adversary can target the viral video content which will be requested by the majority of network users. Thus, the attack can also be subjected to a specific video. In result, all the users requesting that video shall be impacted by the attack if victim and adversary are sharing at least one on-path router towards the source of the video. In particular, the number of nodes negatively impacted by our BOA attack is dependent on the position of the attacker node(s) with respect to the client (benign) nodes. All the client nodes that shares with the attacker, one or more intermediate routers, on the route towards the producer will be affected adversely. The impact will increase as the number of common intermediate routers increases on the route between the client node and producer. Moreover, the client nodes that shares the same first hop router with the attacker will be affected most.

4. A Bitrate Oscillation Attack for DAS over NDN

This section presents the working methodology of our Bitrate Oscillation Attack (BOA) that impacts negatively to the QoE perceived by the NDN consumers (C) during the streaming of multimedia content. The application that performs the streaming process make use of DASH module over NDN. To deliberately reduce the quality of the received multimedia content at C during a streaming session, adversary (Adv) exploits the functionality of the rate adaption function of DASH streaming control system. The Adv launches the attack by inducing frequent and large amplitude bitrate oscillations in the DASH streaming system, which results in frequent switching between highly variable representations (e.g., $b+$ and $b-$) of the multimedia content, as it is shown in Fig. 2. In particular, assume that C needs to watch a video file (say S). To launch the BOA, Adv strategically requests some segments of S before the C starts to fetching it. The segments returned from P through the network to Adv are stored on each intermediate router that earlier forwarded the corresponding interests [6]. Now, at the latter time when C sent the interest packets to fetch the segments of S, some of the these interest requests will be satisfied by P while some from intermediate routers. Due to this, the DASH adaptive system at C will estimate different adaptation rates for alternative segments of the S which leads to the degradation of the QoE perceived by the consumer.

To perform the BOA, we assume that Adv knows about the multimedia content (i.e., S) that the C will be requesting from the network in near future. However, Adv does not know about the resolution(s) in which the C will request the content (S), it is because it depends on the DAS adaptation module running at C. Therefore, Adv fetches the segments in all the available resolutions. Initially, any consumer (like C or Adv) receives the MPD file which has the details about all the segments along with their available bitrates/representations of a video file that the consumer wants to fetch [19]. To successfully perform the BOA, the aim of Adv is to use the information listed in MPD to request a number of segments (in all the available bitrates) of S in a way that leads to frequent and high amplitude oscillations for C. All these segments are cached at intermediate routes during their way to Adv. Recall that, in NDN, when the interest packets traverses towards the P the intermediate routers store a state in their PIT to satisfy the requirements of interest aggregation [3]. When P receives the interest it sends the requested content back into the network, and the content follows the same route through which the interest packets are received.

The Adv issues legitimate interest messages for S in a predefined sequence to ensure the BOA at consumer. As mentioned above that Adv exploits the interest aggregation feature of NDN to store all issued interests at the on-route router's PIT. In our attack model, Adv sent interest messages for the segments of S in ascending order, but it generate interests for alternative segments and for all available bitrates. As each

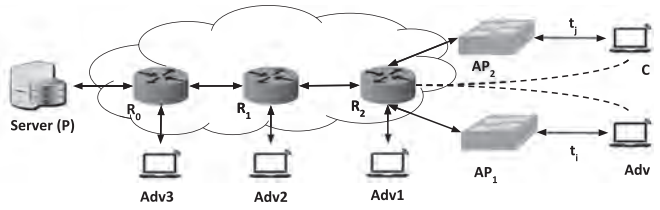

Fig. 3. Topology considered

interest requested by Adv is for new S_n, therefore, every interest makes a PIT entry in the router and is being forwarded the upstream routers. As shown in Fig. 3, later, when C subsequently requests the segments of S (i.e., starts the video streaming), either Adv has previously requested that segment or the segment was one of the segments which is skipped by Adv, i.e., not requested. For instance, take a particular segment say S_n, which is requested by C. If S_n has been previously requested by an Adv and it is received from P, then S_n will be cached at R_2, and it will be provided to C within a round-trip time (RTT) between C and R_2. Moreover, if S_n has been requested by Adv but it is not yet arrived to R_2, then the interest packet sent for S_n from C will be aggregated at R_2 [3]. In this case, the C will receive S_n in less than the RTT between C and P. The RTT depends on the position of S_n on the path from P to Adv. Lastly, if Adv has never requested S_n in the past, then the interest request from C will be forwarded upto P, assuming that S_n is not stored at any intermediate router cache. In this case, the S_n will take the full RTT between C and P to be delivered to C.

When affected by the attack, the C calculates relatively short RTT for segments that are pre-fetched by Adv, and the DASH adaptation logic will assume it as an indication of high available bandwidth in the network. By pre-fetching segments of S with a predefined pattern (e.g., consecutive gaps), the Adv could force the DAS adaptation logic at C to frequently switch between different representations (i.e., very high and low) of S. Next, we discuss the design and functionality of our proposed BOA Algorithm 1 which runs at Adv, i.e., an Adv request content in an

Algorithm 1 Algorithm at Adversary to launch BOA

procedure SEQUENCE_OF_INTEREST (S, i, j, α)
  $MPD \leftarrow Send\_requests\_to\_P$ ▷ $MPD = \{S(n)_{b_{i,j}}\}$
  for $n = 1, n \leqslant N, n + \alpha$ **do**
    $Content(S(n)_b) \leftarrow Interest(S(n)_b)$
    for $i \leqslant f \leqslant j$ **do**
      $Content(S(n)_{b_f}) \leftarrow Interest(S(n)_{b_f})$ ▷ $S(n)_{b_i}, \ldots S(n)_{b_j}$ *caches on* CS_k
    end for
  end for
close;

ascending order with a predefined gap (α). Finally, we will investigate the behavior of C in the presence of our BOA algorithm running at Adv.

At time t_i, the Adv retrieves the MPD and issues a series of interest messages in ascending order (i.e., $S(n+\alpha)$). Here α is the variable or fixed length consecutive gap so that the adversary can issue the discontinuous requests toward P in all the available bitrates ($b_{(i,j)}$) of S. Each request issued by the Adv is for a new segment with all available bitrates ($b_{(i,j)}$) till it requests a total of N number of segments. For each interest, say $S(n+\alpha)_{b_{(i,j)}}$, the router first check the segment in their cache. If not found then check whether it is listed in the PIT, if not, the router forward the interest to FIB [6] in order to forward it to producer. All such forwarded interests traverse the path $Adv \rightarrow Ap_1 \rightarrow R_2 \rightarrow R_1 \rightarrow R_0$. Due to the functioning of NDN [3], the data segments corresponding to interests follows reverse path, and along their way these segment are cached at intermediate routers. Lets assume that at time t_j, C retrieves the MPD for a desired content. As per the information in MPD, C transmit interests to fetch S_n in a bitrate (say b_k). b_k is estimated by DAS adaption logic using the current network conditions [8]. While streaming, when C request a segment by sending the interest $S(n+\alpha)$, the interest will traverse the route $C \rightarrow Ap_2 \rightarrow R_2$. It is because R_2 have the requested content in its cache (CS_2) and it will satisfy the received interest with it.Let B_f be the bitrate representation of the segment received from CS_2, and b_{f+1} be the next, and b_i, b_j be the available lowest and maximum bitrate. As the interest $S(n+\alpha)$ is satisfied from R_2 cache, the C estimates high download rate for the next segment, and accordingly the DASH adaptation unit at C will request the highest resolution (i.e., b_j). That is, C requests $S(n+\alpha+1)$ with b_j, but, the interest will traverse toward P through the path $C \rightarrow Ap_1 \rightarrow R_2 \rightarrow R_1 \rightarrow R_0$ because the requested segment has not yet been cached at any of the intermediate routers. As a result, when C receives the $S(n+\alpha+1)$, its adaption unit calculates low available bandwidth, thus it will estimate a lower network throughput. It is due to the increased RTT of $S(n+\alpha+1)$. Hence, in this way, due to the BOA attack, C will select a lower representation for the next segment. If this process is repeated multiple times by the Adv, then it will lead in alternative cache hit and miss at routers for the interest messages sent by C.

Algorithms 2 and 3 shows the procedure that C uses to stream a video

Algorithm 2 Victim (C) segment selection process

procedure SELECT_SEGMENT_PROC$(S(n)_{b_k}, i, j, \alpha, f)$
  MPD $\leftarrow$ *Send_* requests_ to_ P ▷ $MPD = \{S(n)_{b_{i,j}}\}$
  $S(r)_{b_k} \leftarrow Select\_segment\_proc()$
  $Content(S(r)_{b_k}) \leftarrow Interest(S(r)_{b_k})$
  $S(n)_{b_k} \leftarrow S(r)_{b_k}$
  $ind \leftarrow r$
  while $ind \neq N$ **do**
    $k \leftarrow Bitrate\_adaptation\_proc(S(n)_{b_k}, i, j, \alpha, f)$
    $Content(S(m)_{b_k}) \leftarrow Request(S(m)_{b_k})$ ▷ $m \in (r, N]$
    $S(n)_{b_k} \leftarrow S(m)_{b_k}$
    $ind \leftarrow m$
  end while
close;

Algorithm 3 Victim (C) bitrate selection process

procedure BITRATE_ADAPTATION_PROC$(S(n)_{b_k}, i, j, \alpha, f)$
  if $n == r + q \times \alpha$ **then** ▷ $q \in \{1, \ldots, (N-r)div\ \alpha\}$
    $download\_rate \leftarrow Adaptation\_control\_sys()$
    if $k \leqslant download\_rate$ **then**
      $temp\_k \leftarrow w$ ▷ $w \in (i, j]$
    else
      $temp\_k \leftarrow w$ ▷ $w \in [i, j)$
    end if
  else
    $download\_rate \leftarrow Adaptation\_control\_sys()$
    if $k \geqslant download\_rate$ **then**
      $temp\_k \leftarrow w$ ▷ $w \in i$
    else
      $temp\_k \leftarrow w$ ▷ $w \in (i, j]$
    end if
  end if
  Return $temp\_k$
close;

file (say $S(n)_{b_{i,j}}$) when a BOA attack is ongoing. Algorithm 2 depicts the procedure in which C sends interest messages to request N segments in a sequential order. The selection of bitrate for the requested segments are given in Algorithm 3. The adaptation control logic of DAS calculates the data rate to select the optimal download bitrate for all the segments that it requests. Since the download rate at C depends on the RTT of

previous segment(s) and if the segment has been received from *CS*, then for the segments $S(n+\alpha+1)$, the *C* will request the highest bitrate, i.e., b_j. However, due to low download rate for the subsequent segment, the lower bitrate is will be requested for the next requested segment by *C*.

Due to the above mentioned functionality of procedures, the *C* will experience undesirable bitrate oscillations, manifesting as continuous switches between immensely high and low representations, and it will lead to degradation in user-perceived QoE. Moreover, the playback buffer depletes in case of repeated oscillations, and it will force *C* to take radical measures to refill it at the expense of smooth streaming. Note that default in-network/partial caching feature can also lead to oscillations for a benign user. However, these bitrate oscillations will not lead to a noticeable degradation of QoE at user. This is because in a normal scenario a bunch of continues popular content gets stored on the routers and, each DASH client is experiencing individual network conditions. In result, the oscillation frequency and amplitude will be very low to count as QoE degradation. Moreover, content popularity may lead to a increase in QoE for users in such cases. In contrast, the BoA forces the honest client(s) to stream continuously in immensely different available bitrates (i.e., very high and very low resolutions) while streaming a video file.

5. Network Coding in Name Data Networking

The Network Coding (NC) is an approach in which data delivery to the consumers is made in such a way that prior to being forwarded, the available data packets are combined (i.e., coded) at source(s) and intermediate routers [23,24,39,40]. Therefore, in NC enabled NDN architecture, the network coded data packet comprise evidence from the entire data which have been combined (i.e., coded) to generate them. The fundamental objective of integrating NC in NDN is that consumer no longer requires to request specific data segments. Instead, it requests encoded data segments which also posses similar information.

In NDN, an interest $i_{n,l}$ issued by consumer requests for any specific data packet $s_{n,l}$, where l is the data packet ID. On the other side, in NC enabled NDN, an interest $\hat{i}_n$ is being requested for the network coded data packet $\hat{s}_n$, without mentioning any specific data packet ID, l. Therefore, in order to retrieve $\hat{S}_n$, a set a interests $\hat{I}_n = \{\hat{i}_n\}$ are required. In particular, the set denotes to a single interest. To retrieve the requested content, every consumer sends the interest $\hat{i}_n$ for at least $|S_n|$ times. Note that to retrieve the original content, more than $|S_n|$ data packets might be required since the generation of network coded data packets is by randomly combining the original data packets of a specific name prefix[1]. Also, in case of packet loss, additional interests are required to be issued to recover the lost data packets. Any node receiving network coded interests is able to reply with network coded data packets that match the specified name prefix and are available in the CS.

To be more precise, let's denote the network coded data packet $\hat{s}_n$ as a vector $\hat{\mathbf{s}}_\mathbf{n}$, where each individual element of a vector fits in a finite field. Therefore, the set of including multiple network coded data packets (let say, $|S_n|$) can also be denoted as a matrix $\hat{\mathbf{S}}_\mathbf{n}$, where each row corresponds to a unique data packet $\hat{\mathbf{s}}_\mathbf{n}$. In particular, to produce network coded data packet $\hat{\mathbf{s}}_\mathbf{n}$ any node executes the following operations: $\hat{\mathbf{s}}_\mathbf{n} = \mu.\hat{\mathbf{S}}_\mathbf{n}$, where μ denotes the matrix of coding coefficients which is drawn from a finite field, and $\hat{\mathbf{S}}_\mathbf{n}$ is the matrix made by the set of data packets $\hat{S}_n$. In random linear network coding [41], the selection of coding coefficient μ is random within a large finite field. In result, the data packets generated also have a high probability of being linearly independent to data packets which are being previously generated. In this way, the newly generated data packets are always innovative [41].

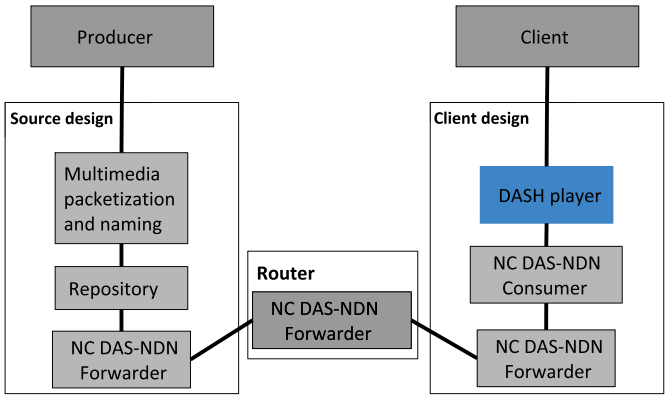

Fig. 4. Proposed NC enabled robust architecture for DAS over NDN

6. NC based DAS-NDN: Network Coding based robust framework for DAS over NDN

In this section, we illustrate the functional and implementation details of proposed BOA countermeasure called *NC based DAS-NDN* that significantly encounters the adverse effects of BOA, and provides a robust architecture for DAS over NDN. In contrast to existing HTTP based DAS, we claim that in NDN, it is not enough to calculate the bitrate for subsequent segments just by taking into account the estimation values[2] of in-line receiving segments. It is due to the fact that NDN enables multipath communication, i.e., the content producer can be reached through multiple paths because of content source variation triggered by NDN's inherent features (i.e., dynamic in-network caching and interest aggregation. This makes DAS estimate enormous and unexpected differences for successive segments retrieved in an on-going video streaming session. In particular, the segment(s) that are retrieved from the intermediate routers creates highly positive estimations as compared to the ones received from the original content producer (*P*). We show (in Section 4) that during a video session, if the variation in the successive segment's location is too frequent, DASH falsely estimates a higher or lower bitrate (i.e., resolution) for the succeeding segments, and it exposes a vulnerability in DAS over NDN.

We claim that in DAS over NDN, it is not satisfactory to discuss multipath streaming for efficient content distribution. To be robust against BOA and varied content source locations, the network should effectively utilize the multimedia content, i.e., cache and route the content based on the optimal resolutions/qualities to reduce bitrate oscillations. Thus it efficiently delivers the best possible perceived QoE. The compulsion specifies that each network node should have a complete and network-wide level view of the content information which relates to the perceived QoE. However, such a solution necessitates to exchange and process enormous amounts of information very frequently. We propose to address this problem by adapting NC for efficient and robust DAS over NDN. Our framework for *NC based DAS-NDN* efficiently combines (i.e., through NC) and forwards the multimedia content based on perceived QoE parameters, thus, reduce bitrate oscillations with varied source locations and multipath transmission.

Below, we present the proposed NC enabled robust architecture for DAS over NDN. In Fig. 4, we illustrate the main components of our NC-enabled NDN architecture, i.e., content producer (*P*), source and client design, and router (*R*). In the following sections, we will discuss the implementation of the nodes in more detail.

[1] As a practical network coding approach, random linear network coding [22,24] is used in the paper.

[2] We refer here to all the parameters utilized by bitrate adaptation strategies, such as bandwidth, buffer occupancy or hybrid, as discussed in Section 3.

6.1. Source Design

We consider an NC enabled content source which acts as a repository of network coded content (say S_{rep}). In particular, S_{rep} when receives the interests can reply with the content which is in the repository. Unlike CS, using persistent storage devices, S_{rep} is able to store the data packets for a longer period of time. Before the procedure of DAS is being started by the clients, S_{rep} receives DASH segments from P, and further break them into data packets with associated names. It is because the network devices use a maximum transmission unit (MTU) while processing an incoming data packet. Therefore, the produced segments are further fragmented into smaller data packets (or chunks) before their transmission. Finally, the data packets are converted into their network coded version and loaded in a repository. After receiving the interests, S_{rep} satisfies interests by delivering the network coded version of the data packets. Below, we discuss the functional details of the source design process.

6.1.1. Fragmentation of Multimedia data packets

Each DASH segment is broken in a number of fragments (called data packets) in a way that each packet $s_n \ \epsilon \ S_n$ fits in the size of MTU. As stated in Section 5, each data packet s_n is represented as a vector $\mathbf{s_n}$. To practically implement NC, an encoding vector is pre-pended with each data packet $\mathbf{s_n}$. This makes both the clients and routers aware of coding operations which are being subjected to network coded data packets, i.e., $\hat{s}_n$. At S_{rep}, the initial value (i.e., $\mathbf{s_{n,v}}$) of encoding vector corresponding to a data packet is set to υth unit vector. The υth place of this vector is set to 1, and the other places are set to 0.

The DASH segments are composed of the immense number of data packets. Therefore, by pre-pending encoding vectors to each packet creates additional communication overhead, which leads to the consumption of network resources. To limit the overhead, we use the idea of *generations* [42]. In particular, the data packets from the original set composing S_n are divided into smaller groups called generations [42]. Then the similar coding operations are implemented on data packets that are part of the same generations [42]. It further reduces the network coding overhead.

At S_{rep}, we design the generations for DASH segments in a way that network coded data version of segments are able to mitigate oscillations caused by BOA and varying content source locations. In particular, each generation overlaps the ascending DASH segments into their respective higher bitrates. Let's assume that the data packets which creates the generation g is denoted as $S_{n,b,g}$, where b is the bitrate and g is the generation ID[3]. Thus, $S_n = \cup_{g=1}^{G} S_{n,b,g}$, where G denotes the total number of generations which compose $S_{n,b}$. To avoid the mixing of data packets that belongs to different generations, we append the generation ID (g) to the name of a data packet (i.e., URI) such that `/video/100kbit/bunny_2s1.m4s/generation_ID`.

In NC based DAS-NDN, we implement the generations such that coding operation is expanded to all the data packets belonging to two successive segments and their respective higher bitrates, i.e., the generation size is expanded to the consecutive segments and their optimal resolutions. In particular, each generation g contains at least some of the data packets which belongs to two consecutive segments in ascending order and their respective higher bitrates. This is driven by the fact that DASH clients utilize the estimation of the received segment when selecting the bitrate of the subsequent segment. This way, generations forward and cache the multimedia content that avoids bitrate oscillations triggered by BOA and varying content source locations.

Fig. 5 illustrates the mapping of two generations, i.e., g and $g+1$, respectively, on original data packets. Here two consecutive segments $S1(b_i)$ and $S2(b_j)$ are first decomposed into data packets (i.e., chunks of MTU size), where $j > i$. These data packets are then encoded into network coded data packets. The network coded data packets are the random linear combination of original data packets. Fig. 5 also shows the that in generation g, the data packets of bitrates i of $S1$ are encoded with bitrates j of $S2$. Thus, more encoded packets of $S2$ with higher representations (i.e., j) are required to decoded the original segment of $S1_{b,i}$.

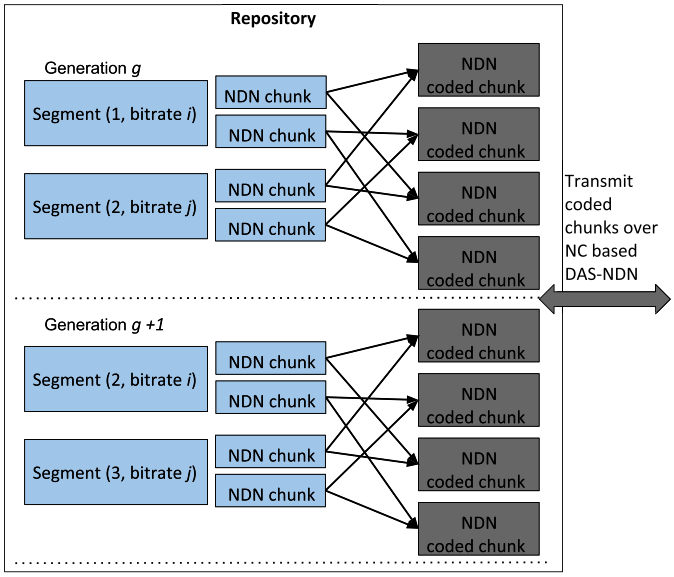

Fig. 5. Multimedia data packetization and mapping of generations

With the above-described generation mapping of original data to form a network coded version of data enables the NDN caching and routing strategies to enable network-wide goals for DASH. In particular, in case, a request from the DASH client results in a cache-hit, the CS or S_{rp} provides the user encoded data which is capable of competing the DASH bandwidth estimation requirements for the subsequent segment.

6.1.2. Repository

Each S_{rp} contains one or more repositories which contain network coded version of original data. Note that the repository may not contain the complete set of data packets, i.e., $S_{n,b,g}$, but only the subset of data packets such as $S^{u}_{n,b,g} \ \epsilon \ |S_{n,b,g}|$. In addition, we consider that the functionalities of repository and *CS* are similar such as: (i) on receiving the data packets, these are cached, and (ii) on receiving the interest $\hat{i}_{n,b}$ against network coded packets, it replies with coded data packets, i.e., $\hat{s}_{n,b,g}$, which are generated by a set of data packets ($S_{n,b,g}$) available at CS repository.

6.2. Client Design

In NC based DAS-NDN architecture, the NDN client consists of two fundamental components: (i) a DASH streaming control player which adapts the user requests based on available bitrates and network bandwidth to provide best possible perceived QoE, and (ii) an NC based DAS-NDN consumer that receives requests of original DASH segments from the end-user and generates network coded interests to retrieve network coded data packets. Below we present the details of both:

6.2.1. The DASH Player

The DASH player is the most immediate interface between the end-user and proposed video communication model. When the DASH player receives a request from end user to retrieve a video (S), initially, it requests the MPD file which defines the relationship between a video segment's associated characteristics and its name. The MPD file is usually small in size and it needs to be communicated only once. Thus, it is not network coded and is requested as traditional NDN content object. Each MPD lists the NDN names (i.e., Uniform Resource Identifier) of

[3] The generation ID restricts the data packets to not mix with the data packets belonging to other generations

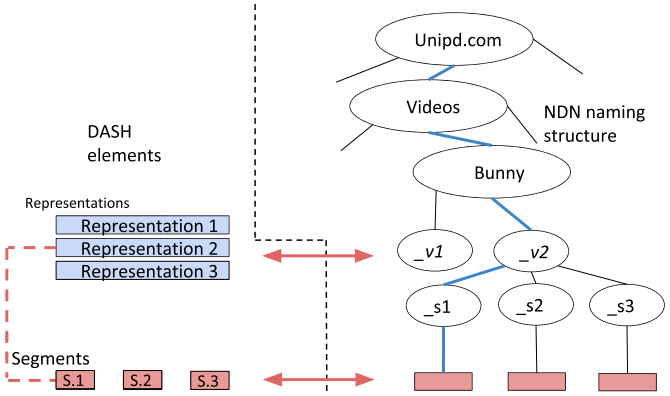

Fig. 6. NDN's naming scheme for DASH's content representation

the video segments [12]. The NDN naming scheme in DAS over NDN supports versioning and segmentation, which is used for video streaming [16]. The versioning of segments in NDN indicate different representations of DASH-based multimedia content. Fig. 6 illustrates the example of NDN name versioning, which is chosen to map different representations of DASH segments [16]. For instance, the *representation* 1 and 2 are directed to the versions of NDN URI denoted by $\_v1$ and $\_v2$, respectively. Similarly, DASH segmentation structure for the content is also supported by the NDN naming scheme. Given the MPD information, a DASH player can request the segment (S_n) appropriate to the estimation of the available bandwidth and network conditions [26].

6.2.2. The NC based DAS-NDN Consumer

When network coded data packets are retrieved at NC based DAS-NDN consumer, it first decodes them and then reassembles them to the original DASH segment. This process is done before the segment has been sent to DASH streaming control system. In particular, when a consumer receives a request for a DASH segment $S_{n,b}$, it generate and send a set of network coded interests (i.e.,$\hat{I}_{n,b,g}$), here n and b are the segment's version and resolution, and g is the generation ID. Note that the generation ID starts from $g = 1$. Please note that the value of G can also be obtained through an initial meta-data packet.

To keeps track of all the received innovative data packets $\hat{s}_{n,b,g}$, NC based DAS-NDN consumer uses the matrix of data packets $\hat{S}_{n,b,g}$. When the matrix ($\hat{S}_{n,b,g}$) is in full rank, and the Gaussian elimination is used to retrieve original data packets [42].

6.3. The NC based DAS-NDN forwarder

The key role of NC based DAS-NDN forwarder in our architecture is as follows: (i) first, it routes the network coded interests toward their content sources, (ii) it uses the reverse path [3] to forward the corresponding NC data packets back to the clients, (iii) it applies network coding operations before forwarding the packets, and (iv) it caches the forwarded packets to satisfy similar future requests. Below we present the key components of NC based DAS-NDN forwarder, i.e., a modified version of CS, a PIT and a FIB. The FIB of NC based DAS-NDN works as similar to the FIB of traditional NDN architecture [3].

6.3.1. Content Store

The NDN routers maintain a state in their cache for data packets that are earlier received. It is done to reply the future interest requests (let say, i_n) which holds the same name prefix of cached content (e.g., s_n). Differently from NDN, in NC based DAS-NDN, the clients provide a name prefix ($\hat{i}_{n,b,g}$) which refer to a NC data packet that is generated from the set $\hat{S}_{n,b,g}$ [42], along with the precise name for the interests. Note that g is the generation ID. In particular, in NC based DAS-NDN, a router R receiving an interest $\hat{i}_{n,b,g}$ reply with the NC version of the data packet $\hat{s}_{n,b,g}$. However, R only replies to $\hat{i}_{n,b,g}$ to the client if it identifies that $\hat{s}_{n,b,g}$ will have high probability of being innovative. To generate innovative data packets (i.e., $\hat{s}_{n,b,g}$), R randomly combines the data packets $\hat{\mathbf{S}}_{\mathbf{n,b,g}}$ in its CS [42]. Thus, $\hat{s}_{n,b,g}$ can be expressed as $\hat{s}_{n,b,g} = \sum_{q=1}^{|\hat{\mathbf{S}}_{\mathbf{n,b,g}}|} \mu_q * \hat{\mathbf{s}}^{(\mathbf{q})}_{\mathbf{n,b,g}}$, here the randomly selected coding coefficient is μ_q and $\hat{\mathbf{s}}^{(\mathbf{q})}_{\mathbf{n,b,g}}$ is the qth data packet in $\hat{\mathbf{S}}_{\mathbf{n,b,g}}$.

6.3.2. Pending Interest Table

As stated in Section 6.3.1, upon receiving an interest $\hat{i}_{n,b,g}$, the NC based DAS-NDN router R replies such interest with NC data packet which is generated from $\hat{S}_{n,b,g}$. In case, the data packets which are stored in R's CS are not enough to create innovative packets $\hat{s}_{n,b,g}$, then R waits for new packets to receive before it replies to such an interest. Moreover, the R forward the interest $\hat{i}_{n,b,g}$ to the next hop and creates a PIT entry for that interest similar to NDN [3]. To facilitate the additional functionalities of NC based DAS-NDN forwarder, the PIT entry also includes the generation g along with name prefix, e.g., a list of PIT entries at R is expressed as $\tau = \{t_{n,b,g} \ldots\}$ [42].

6.4. Mitigation Against BOA

In this section, we present the functional details of NC based DAS-NDN mitigation mechanism against BOA. In particular, we illustrate the procedure when our proposed NC enabled network (i.e., NC based DAS-NDN router) receives a interest from a benign DASH client and the data packet processing at a client in the presence of BOA attack or varied content source locations.

6.4.1. Interest processing

When a NC based DAS-NDN router receive a request $\hat{i}_{n,b,g}$ for network coded data packet[4] $\hat{s}_{n,b,g}$, the router performs one of the function from the following three conditions: (i) satisfies the interest with NC data packet generated from a set of data packets available at R's CS, (ii) forwards that interest to next-hop to retrieve innovative NC data packets, or (iii) interest collapse in the PIT and waits for the new coded data packet to arrive. We explain this procedure in detail in Algorithm 4.

- Satisfying the interest: Router R replies to interest $\hat{i}_{n,b,g}$ when (i) it has collected innovative network coded data packets $|\hat{S}_{n,b,g}|$ in its CS, i.e., data packets are available to decode generation for $\hat{s}_{n,b,g}$; or (ii) the network coded data packets generated by R have a high probability of being innovative for the requesting node. The total number of coded data packets which R is able to generate over the face f, and which also have the probability of being innovative is calculated by $\epsilon^{f}_{n,b,g} = rank(\hat{S}^{R}_{n,b,g}) - \sigma^{f}_{n,b,g}$. The parameter $\sigma^{f}_{n,b,g}$ denotes the number of network coded data packets that have been forwarded over the face f. In particular, when the $\epsilon^{f}_{n,b,g}$ is greater than 0, the router R generates a new network coded data packets and forward it to face f.
  As stated in Section 6.1.1, the generations g for each DASH segment $S_{n,b_i,g}$ are packetize along with their subsequent segments in higher bitrates $S_{n+1,b_j,g}$, where $j > i$. In particular, each segment of multimedia content is network coded along with the subsequent segment and their respective higher bitrates. Therefore, R replies to the interest $\hat{i}_{n,b,g}$ only if the data packet is composed of the content $\hat{S}_{n,b,g}$, which is generated (i.e., coded) with subsequent segments of higher bitrates.
- Forwarding the interest: In case, when R is not able to generate enough network coded data packets or the packets do not have high probability of being innovative, i.e, $\epsilon^{f}_{n,b,g}$ is equal to 0, R forwards the interest $\hat{i}_{n,b,g}$ to next hop. In particular, this means that R requires to

[4] We designed an additional field for `NetworkCodingEnable` along with interest. The field is set to be 1 or 0, respectively, indicating that interest is for network coded data packet or for non-coded data packet.

Algorithm 4 Network Coding enabled DAS

1: **procedure** SELECT_SEGMENT_PROC($S(n)_{b_k}, i, j, f, \hat{i}(n)_{b,g}, \hat{S}(n)_{b,g}$)
2: $MPD \leftarrow Send\_requests\_to\_P$ ▷ $MPD = \{S(n)_{b_{i,j}}\}$
3: $S(r)_{b_k} \leftarrow Select\_segment\_proc()$
4: $\hat{i}(n)_{b_k,g} \leftarrow Interest(S(r)_{b_k})$
5: $\hat{S}(n)_{b_k,g} \leftarrow \hat{i}(r)_{b_k}, g)$ ▷ Issue network coded interests for network coded segments
6: $\hat{S}(r)b_k, g \leftarrow NetCodNDN\_Forwarder\_proc(\hat{i}(r)_{b_k}, g, i, j, f)$ ▷ NetCodNDN forwarder
7: **if** $rank(\hat{S}(r), b, g) = |\hat{S}(r), g|$ **then**
8: *generation g is decodable*
9: $g = S(r)b_i + S(r)b_k$ ▷ $i < k \leq j$
10: *Reply with network coded segment*
11:
12: $S(n)_{b_k} \leftarrow \hat{(}S(r)_{b_k}, g)$
13: **else**
14: **if** $|\hat{S}(r)k, g| \neq innovative\_data\_packets$ **then**
15: *Send interest* $\hat{i}(r)_{b_k}, g)$ *to next hop*
16: **else**
17: **if** $t_{out} > t_{in}$ **then** ▷ PIT have an identical entry
18: *wait for network coded data packets*
19: **end if**
20: **end if**
21: **end if**
22: **close**;

retrieve more innovative data packets which increases the rank of $\mathbf{S_{n,b,g}}$ to reply the interest $\hat{i}_{n,b,g}$. Thus, when R is not able to generate complete coded data for $S_{n,b,g}$ which is composed of data packets associated with $S_{n,b_i,g}$ and $S_{n+1,b_j,g}$, where $j > i$, R do not satisfy the interest and forward it to next hop. Note that before forwarding the interest, R checks the PIT to support interest aggregation feature as stated in Section 6.3.

- Waiting for new network coded data packets: If there is a similar PIT entry for any specific network coded interest, R do not forward the interest, and it will wait to retrieve new network coded packet. In particular, it means that network coded interest sent by R is greater than the data packets received. Therefore, the router waits for new network coded data packets so that it can satisfy all pending interests in the PIT.

6.4.2. Processing of data packets at DASH client

With above-described functionality of NC based DAS-NDN forwarder, DASH client always receive the segments which are linearly network coded with their subsequent segments of higher bitrates. As mentioned in the Algorithm 4 (line 6 - 22), router R while receiving the requests for S_{n,b_i} from DASH client replies with $\hat{S}_{n,b_k,g}$ (i.e., packets coded in the combination of segments, $\hat{S}_{n,b_i,g}$ and $\hat{S}_{n+1,b_k,g}$, where $i > k \leq j$. In case, when DASH client experiences undesirable bitrate estimations due to BOA or varying content source locations, it requests for higher bitrates for subsequent segments. Since the packets are linearly coded with the higher bitrates and are being transmitted over the network through NC enabled forwarding and caching (as illustrated in Section 6.3), the successive coded segments with optimal bitrates are also delivered to the client. Finally, the DASH client retrieves the original segments from the available network coded data packets.

7. Performance Evaluation and Analysis

In this section, we present the evaluation of our proposed NC based DAS-NDN in presence of BOA, and compare with DAS over NDN without network coding capabilities. To examine the impact of attack, we perform extensive simulations on AMuSt-ndnSIM [25], which is an ndnSIM [43] based adaptive video streaming framework. In particular, AMuSt-ndnSIM provides a support to connect libdash[5] [44] and NDN by exchanging HTTP traffic with NDN. To enable network coding for DAS between sources and clients, we modified Amust-ndnSIM framework. In particular, we have used Kodo C++ library [45] to enable network coding operations (as discussed in Section 6) in DAS over NDN..

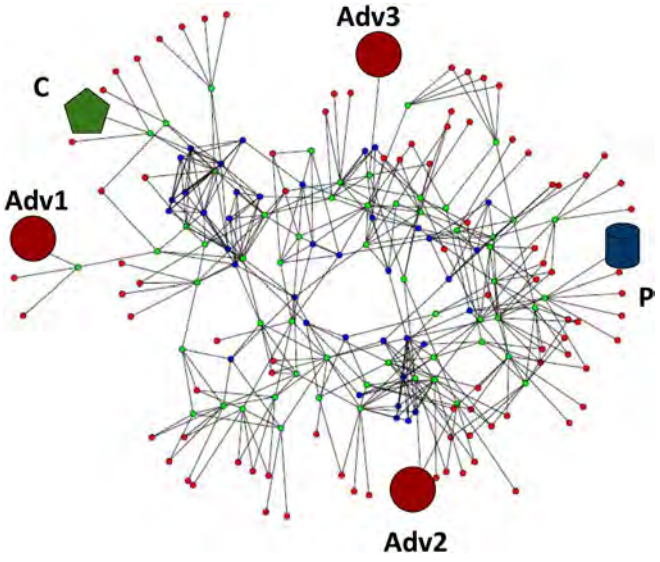

Fig. 7. Internet like topology AS 3967

7.1. Test Setup

We evaluate our proposed NC based DAS-NDN architecture by implementing a real ISP-like topology (i.e., AS-3967) which is measured by the RocketFuel project [46], as illustrated in Fig. 7. The ISP-like topology we used is based on a modified version of Rocketfuels AT&T topology [46]. Aim to select a larger ISP-like topology is to reflect the impact of BOA and effectiveness of proposed countermeasure under a more realistic and large-scale network topology. We extracted the largest connected component comprising of 539 nodes from this original topology and separated the nodes into three categories: clients, gateways, and backbones. Nodes having degree less than four were classified as clients (313 red nodes as shown in Fig. 7), nodes directly connected to clients were classified as gateways (79 green nodes), and the remaining nodes were classified as backbones (147 blue nodes). On AS-3967, we implemented a server P, DASH client C, and proposed adversarial model (i.e., with three adversaries, $Adv1$, $Adv2$, and $Adv3$ at different locations) linked with 79 nodes and 147 bidirectional edges. We configured P to host real-time multimedia data, an MPEG-DASH video (i.e., *BigBuckBunny*) in two variants of data coding such as AVC [30] and SVC-encoded [29] dataset, respectively. We examine C using three different types of DASH adaptation strategies, i.e., Rate-Based (*RB*) [25], Buffer-Based (*BB*) [33][6], and Rate-Buffer-based (*R&B*) [25].

To examine the adversarial impact with diverse conditions, we perform simulations by connecting adversary separately to multiple locations in the network of AS-3967 (as Shown in Fig. 7). Lastly, to prove the authenticity of results we perform each simulation round for a considerable length of time period, i.e., more than four minutes. The details about the network parameters along with their associated values that are used in target setup are depicted in Table 2.

[5] Libdash is an official library providing DASH standards.

[6] We use *Bandwidth independent Efficient Buffering* (BiEB) [33] with a maximum buffer limit of 33 seconds.

Table 1
Notation table

Notation	Meaning
Adv, C, P	adversary, client, producer
R, AP	routers, access point
S, N	video file at P, number of segments in S
S_n	n^{th} segment of S
CS_l	cache at R_l
$b_{(i,j)}$	set of available bit rates of S
α	consecutive gap of variable length
b_f	bitrate of S received from CS
b_j and b_i	maximum and minimum bitrate
MPD	media presentation description (*XML* file)

Table 2
Simulation Setup

Parameter	Value
Total number of DASH segments (N)	250
Simulation time	240
Available representations in AVC	20
SVC quality support layers	4
Each segment duration	2 sec
Maximum buffer size	30
Gap in segments fetched by Adv α	2
Fragment size	1449 byte
Drop Tail Queue (max. packets)	20
Caching policy	LRU
Start-up delay (s)	0.1
Maximum buffered time	30 Sec

7.2. Evaluation Metrics

A video file viewed by a user is considered of a good quality if it is received in high bitrate (i.e., high resolution) and there were no interruptions during the streaming process (i.e., resolution should be stable during the streaming). In the literature [27] and [28], the metrics that have highest impact on user experience during DASH video streaming depends on the frequent switching between bitrates and switching amplitude during the streaming of a video file. Therefore, the spatial quality of a streaming video is calculated by measuring the amplitude and frequency of the variations occurred. To this end, we use the following evaluation metrics.

- Number of oscillations/switches: It is measured as a total number of frequency fluctuations over a time period, higher the number lower the perceived video quality [47];
- Average oscillation/switch magnitude: It is as the total amplitude of the frequency fluctuations over a time period to the number of fluctuations [28,47].

7.3. Attack Impact

To relax the adversarial assumption, we computed the impact of attack when adversary is not completely aware of the victim location. In particular, we launched the attack while connecting adversary separately at different edge points of topology (i.e., AS- 3967). Figs. 8 and 9 show the adversarial impact of BOA separately with different adversarial locations (as shown in Fig. 7). Fig. 8 reports similar variation in the bitrate requested by DAS client for three different adversaries located at various positions in network. Furthermore, results in Fig. 9 shows a merely similar increase in the oscillation frequency and average oscillation magnitude encountered by C for all three cases. In particular, the two figures confirm that Adv is able to degrade the QoE of benign users while being connected to any on-path router. This is due to the fact that in all three cases, the Adv is able to populate the PIT and CS of atleast some on-path routers (i.e., between C and P) with intended requests/content. In results, C while intercepting these states of PIT and

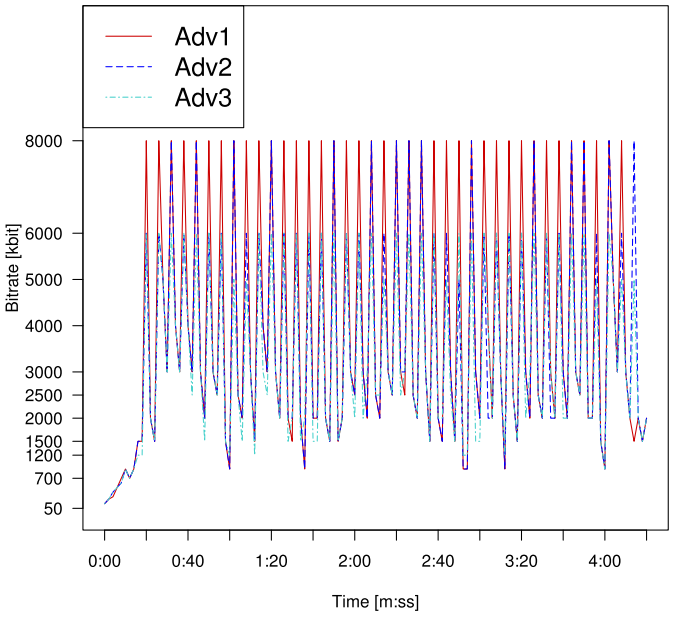

Fig. 8. Bitrate variations with different adversarial locations using RB (AVC)

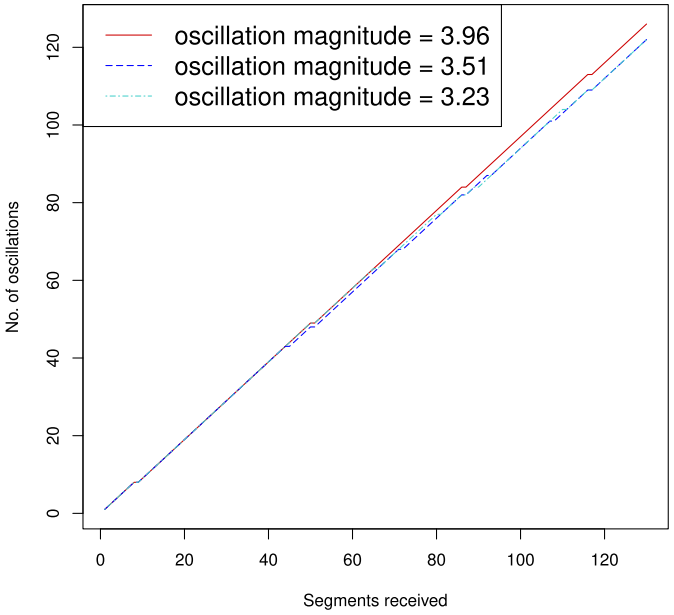

Fig. 9. # of oscillations and average oscillation magnitude to different adversarial locations using RB (AVC)

CS of on-path routers experience bitrate oscillations. Due to space limitations, we present rest of the results including all various adaptation strategies and dataset for the case of $Adv1$ as a benchmark in our simulations.

The experimental investigation also shows that maximum bitrate oscillations during the attack happens when an Adv request interests with consecutive gaps. In case when an Adv issues two contiguous interests (instead of one) with a consecutive gap, the impact of the attack declines by 50%. Moreover, the adversary helps the victim to improve its QoE if it requests more than two continuous segments. Since in this case the client will be receiving most of the continuous segments from the routers which are targeted by Adv (i.e., on-path), it will result in reduced bitrate oscillations and high bandwidth utilization. Therefore in our experimental analysis, the adversary requests a unilateral sequence of segments to achieve the maximum efficiency in the attack. The Figs. 10 and 11 show

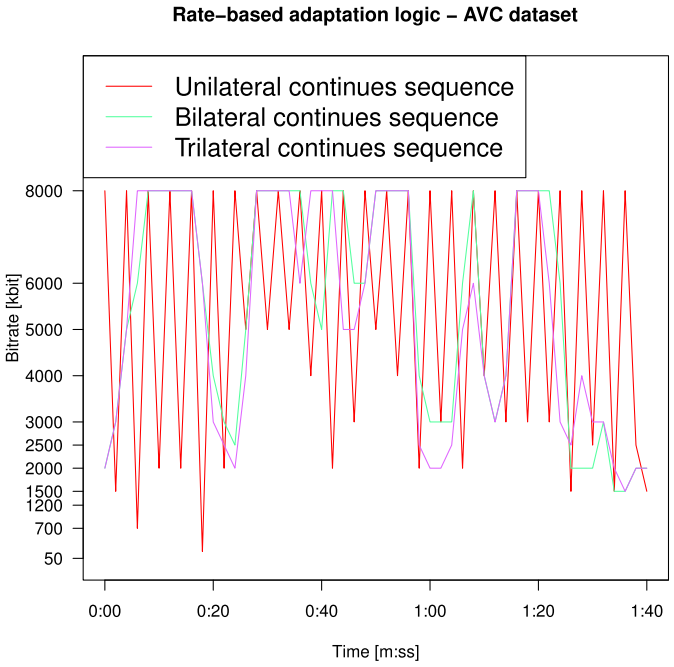

Fig. 10. Comparison of attack sequences (RB-AVC)

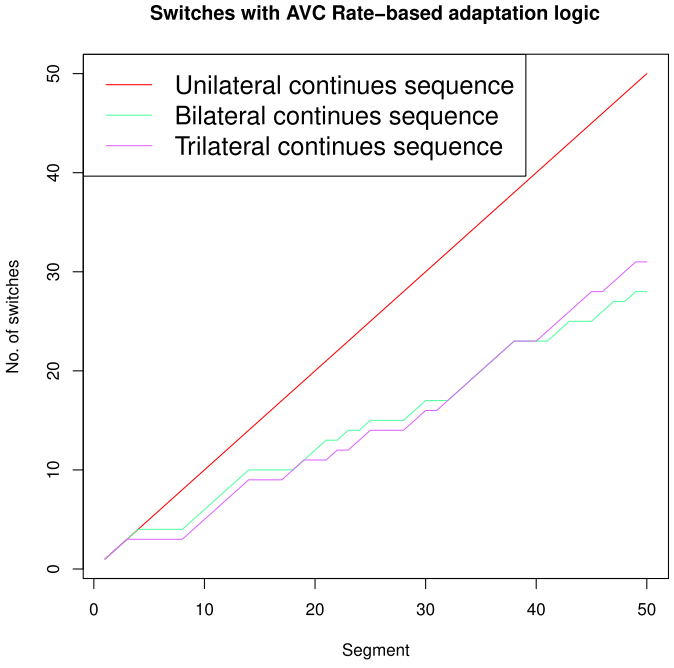

Fig. 11. Switches comparison of attack sequences (RB-AVC)

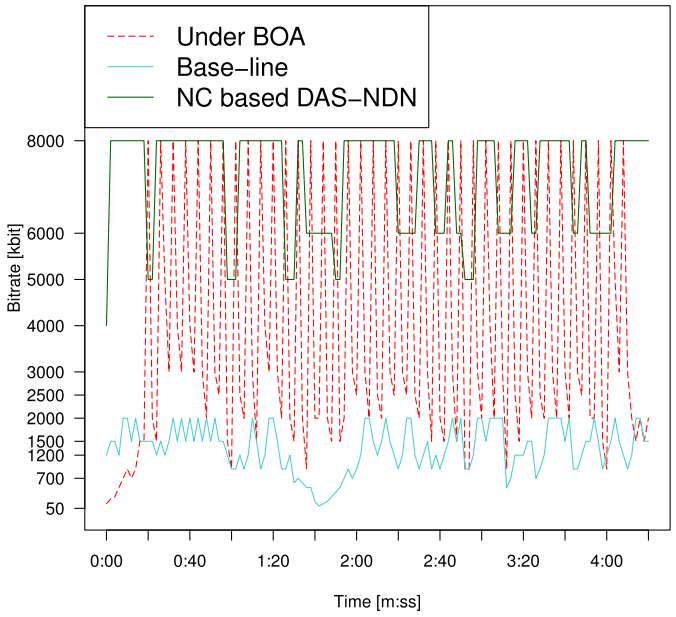

Fig. 12. Dynamic adaptive streaming using RB (AVC)

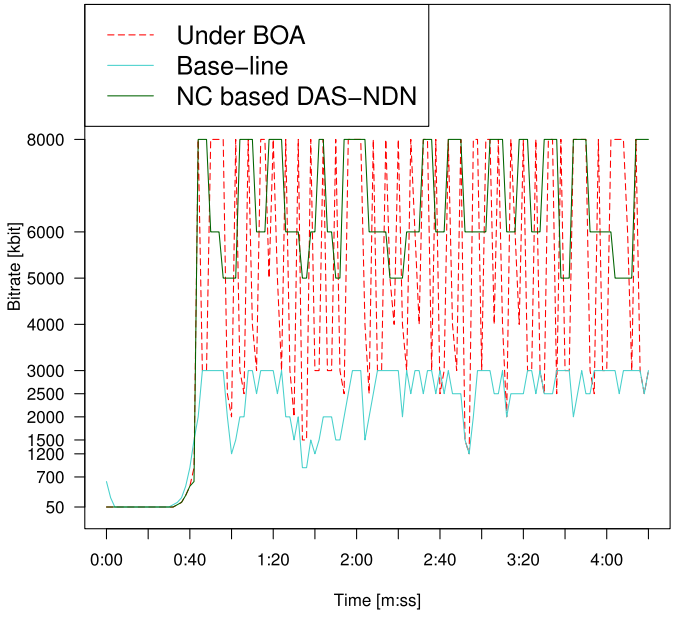

Fig. 13. Dynamic adaptive streaming using R&B (AVC)

the relation of switching frequency and number of continuous segments requested by the attacker. From the results, it can be seen that switching frequency reduces when Adv requests more number of consecutive segments.

To analyze the adversarial impact of BOA, we compare the *base-line* scenario (i.e., without attack) with the scenario where DAS client experience BOA. Using AVC dataset, the bitrate requested by the DAS client using RB and R&B adaptation logic for both the cases, i.e., Under BOA and base-line, and it is shown in Figs. 12 and 13, respectively. Moreover, Figs. 15 and 16 plot the corresponding oscillation frequency experienced by the DAS client. The results show that the frequency of oscillations while streaming with RB and R&B logic increases enormously, i.e., approximately 20% and 33.3%, respectively. In addition, Fig. 22 show the average oscillation magnitude of bitrate fluctuations for these two adaptation logic. The results report that BOA also increases the oscillation magnitude for RB and R&B by upto 267% and 212%, respectively.

The bitrate requested by client using BB adaptation logic (for AVC dataset) is reported in Fig. 14. At first glance, it can be seen that the client is experiencing higher video bitrates under BOA, however, but when considering the user's QoE evaluation metrics (detailed in Section 7.2), it is not the only factor to be acceptable. Results in Fig. 17 reports a visible increase in the oscillation frequency for BB adaptation logic (i.e., up to 33%). This is because the BOA rapidly saturates buffer occupancy for repeated time intervals which results in bitrate oscillations. In addition, Fig. 22 reports the average oscillation magnitude for BB adaption logic that also increases upto 30% under BOA. In summary, for all the DAS adaptation strategies in AVC, there is an increase in num-

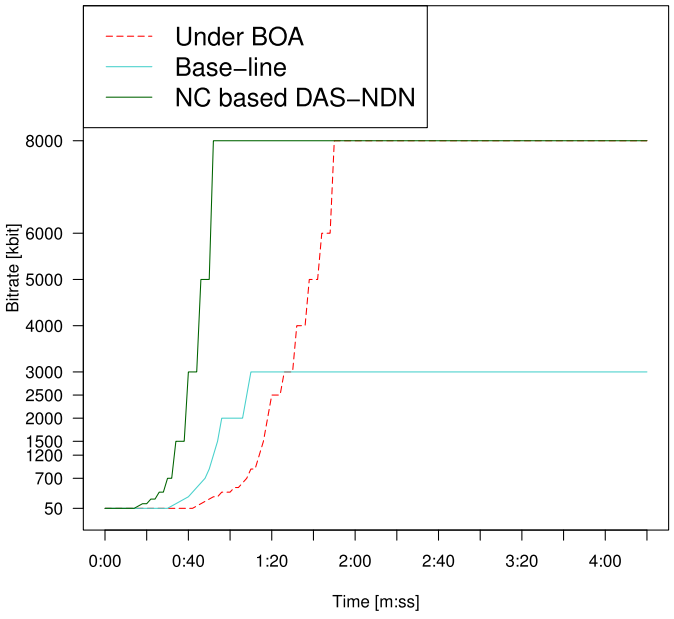

Fig. 14. Dynamic adaptive streaming using BB (AVC)

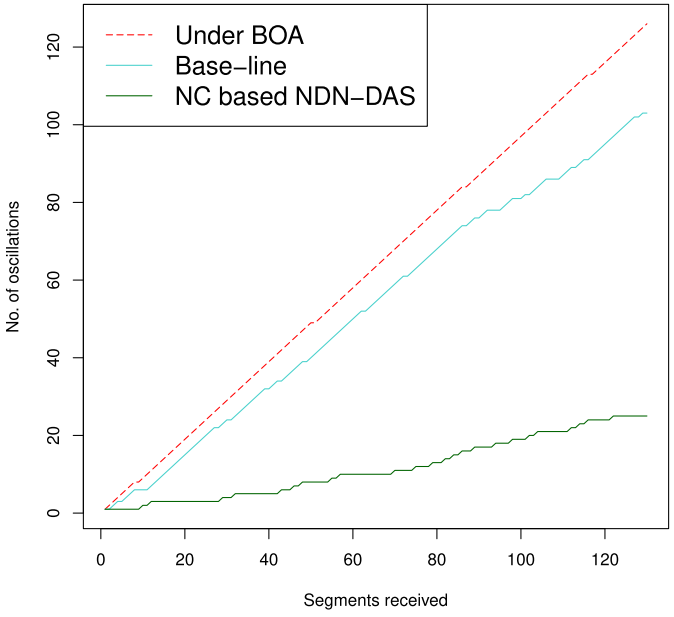

Fig. 15. # of switches RB (AVC)

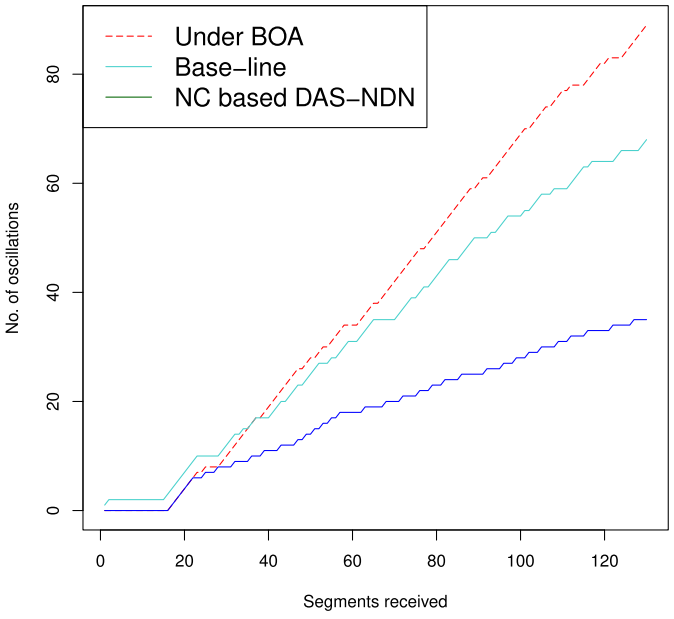

Fig. 16. # of switches R&B (AVC)

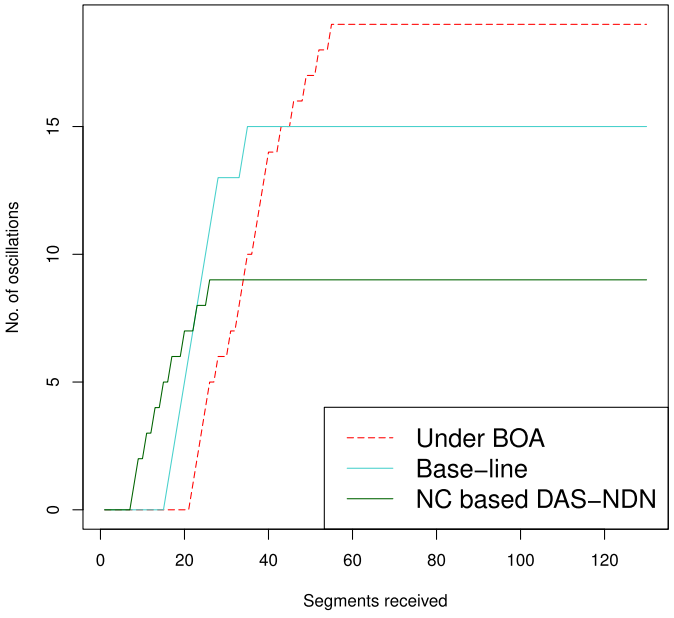

Fig. 17. # of switches BB (AVC)

ber and magnitude of oscillations that the consumer experiences. It is due to the BOA in which the adversary forces the DAS adaptation unit to switch multiple times with extremely low and high bitrates.

For SVC dataset, the BOA results shows the increase in oscillation frequency for RB as it is seen in Figs. 18 and 20, which is about 275% approximately. In addition, Fig. 22 also depicts that oscillation magnitude increases by about 270%, which translates into massive QoE degradation. Note that the overall oscillation magnitude for SVC is less when compared to AVC. It is because, in SVC, the total available bitrate representations are four (i.e., one BL and three EL) as compared to twenty bitrate representations of AVC. Our simulation studies also revealed that BB adaptation logic in SVC is resilient to BOA. Figs. 19 and 21 show that buffer capacity resists to short term bandwidth fluctuations, where there are just few layers of quality and each layer subsequently enhances the video quality. However, use of BB explicitly with SVC still remains an open question for researchers due to buffer size management in small devices, such as smartphones.

7.4. NC based NDN-DAS Effectiveness

In this section, we discuss the effectiveness of proposed countermeasure. Our simulation studies highlight the phenomena that in NC based DAS-NDN, each DASH segment and it's respective representation is mapped through generations with successive segment and respective higher bitrates, as detailed in Section 6.1.1. Therefore, when victim experiences abrupt variation in bitrate estimation and request for higher bitrate (i.e., in case of BOA or varying content source location), the successive segment is also forwarded to end user. This because segment's packets are linearly encoded with their respective higher bitrates of subsequent segments and are being delivered through NC enable forwarding and caching, as illustrated in Section 6.3. It is worth mentioning that NC

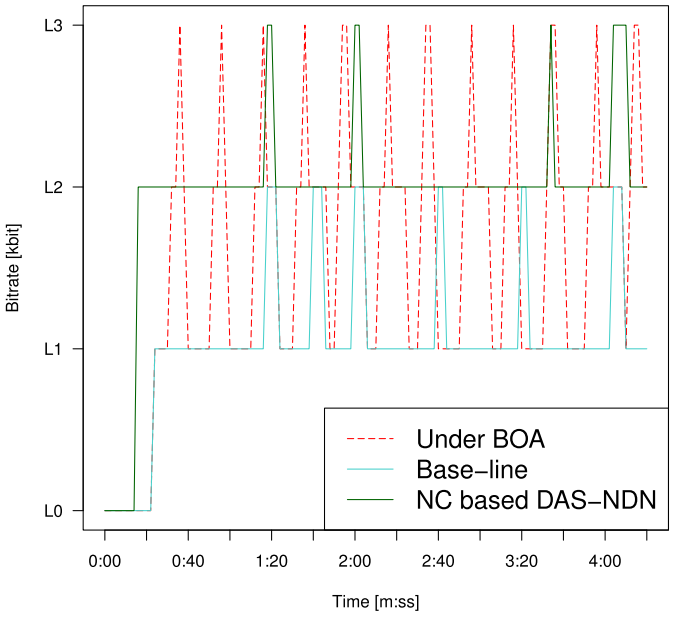

Fig. 18. Dynamic adaptive streaming using RB (SVC)

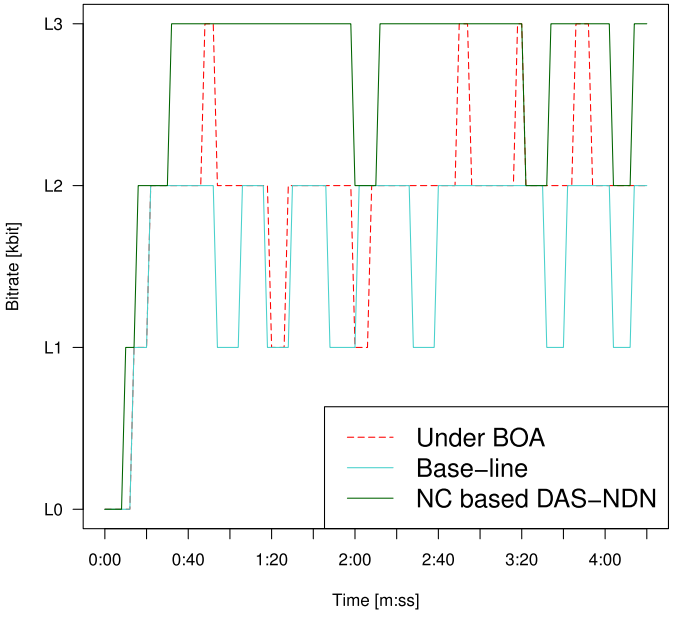

Fig. 19. Dynamic adaptive streaming using BB (SVC)

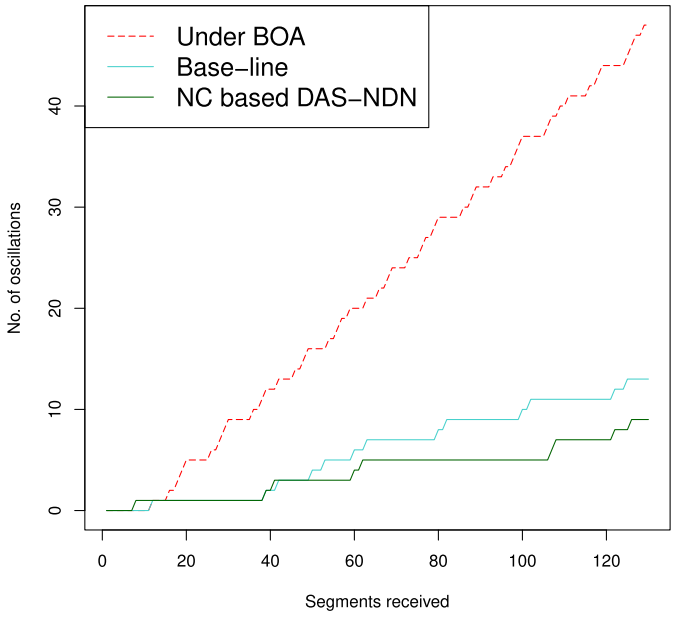

Fig. 20. # of switches RB (SVC)

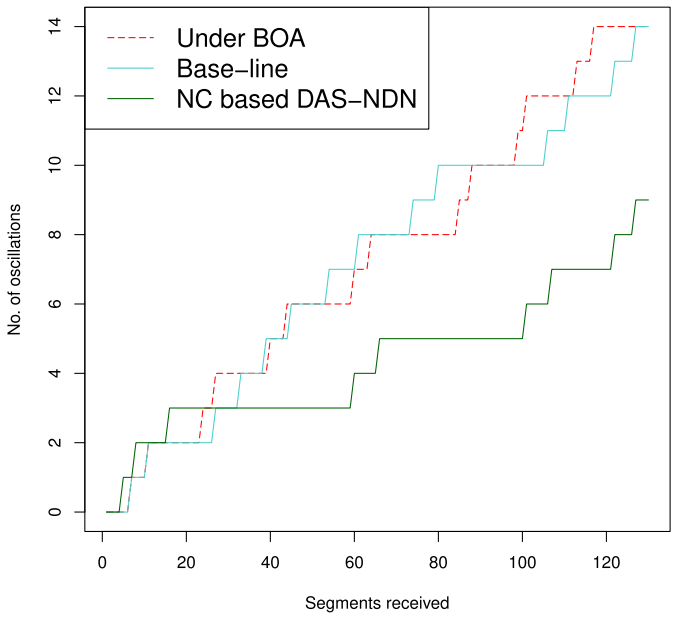

Fig. 21. # of switches BB (SVC)

based DAS-NDN client expediences higher video bitrates while streaming compared to convectional DAS-NDN client and hence improved the perceived QoE.

The simulation results in Figs. 12, 13 and 14 report the performance of NC based DAS-NDN using AVC data set for RB, R&B, and BB, respectively. Results in Figs. 15, 16, and 17 show that our proposed countermeasure not only mitigate the adverse impact of BOA, but it also significantly reduces the oscillation frequency compare to the base-line scenario. In particular, compare to the base-line, NC based DAS-NDN reduces the oscillation frequency by up to 80%, 68%, and 46% for RB, R&B, and BB, respectively. In addition, Fig. 22 reports that NC based DAS-NDN also reduces the magnitude of bitrate fluctuations when compared to base-line, i.e., by up to 275%, 46%, and 17% for RB, R&B, and BB, respectively.

In case of SVC dataset, Figs. 18 and 20 report the performance of NC based DAS-NDN while using RB adaptation strategy. Results report that the proposed countermeasure efficiently mitigates the attack, moreover, compare to base-line reduces the oscillation frequency by up to 20%. Fig. 22 also shows that NC based DAS-NDN reduces oscillation magnitude by to 30% when compared to conventional DAS streaming without adversary. Although simulation studies reveals that BB based DAS adaptation (for SVC) is unaffected by BOA (refer to Figs. 19, 21 and 22), network coding enabled DAS enhances the QoE of user by further reducing the oscillation frequency and magnitude by up to 35% and 40%, respectively.

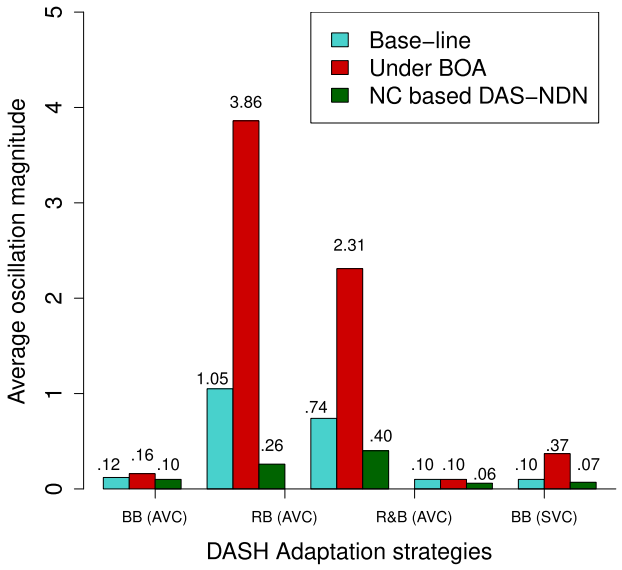

Fig. 22. Average switch magnitude

8. Conclusion

The inherent characteristics of NDN which includes efficient content distribution and tendency to support multipath transmission also bring unexpected privacy consequences [36]. In this paper, first, we show that an adversary can misuse the two absolute specialities of NDN (i.e., in-network caching and interest aggregation) to malfunction the adaptive streaming mechanism of DASH, thus degrades the performance of DAS over NDN. Later, we introduce a robust framework for adaptive bitrate streaming by efficiently enabling network coding to DAS over NDN which mitigates such an attack. To validate our work, we design and implement our proposed approaches (the attack and its countermeasure) on AmustndnSIM simulator. Through extensive simulations, we conclude that the BOA increases the annoyance factor in the user's spatial dimension, i.e., high frequency of bitrate oscillations and oscillation magnitude decrease the perceived QoE. While our proposed countermeasure enables the network coding within NDN architecture w.r.t the perceived QoE evaluation parameters. The results show that the countermeasure effectively alleviates the adverser effects of BOA, moreover, it further heightens user-perceived QoE in the presence of varied content source locations and NDN's implicit characteristics.

Declaration of Competing Interest

The authors declare that there is no conflict of interest of any kind for the submitted manuscript.

CRediT authorship contribution statement

Muhammad Hassan: Conceptualization, Writing - original draft, Methodology, Software. **Mauro Conti:** Supervision, Writing - review & editing. **Chhagan Lal:** Writing - original draft, Visualization, Investigation.

Acknowlgedgments

This work is supported in part by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735. The work of M. Conti was supported under the frameork of "Supporting Talent in ReSearch @ University of Padova STARS Grants program for individual research (STARS@UNIPD 2019)".

References

[1] White paper: Cisco Visual Networking Index (VNI): Forecast and methodology, 2017—2022, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.
[2] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, J. Cao, Named data networking: A survey, Computer Science Review 19 (2016) 15–55, doi:10.1016/j.cosrev.2016.01.001.
[3] L. Zhang, et al., Named data networking, ACM SIGCOMM CCR 44 (3) (2014) 66–73.
[4] M.M.S. Soniya, K. Kumar, A survey on named data networking, in: 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015, pp. 1515–1519, doi:10.1109/ECS.2015.7124841.
[5] Congestion control in named data networking - a survey, 86, 2016, pp. 1–11, doi:10.1016/j.comcom.2016.04.017.
[6] V. Jacobson, et al., Networking named content, in: ACM International Conference on Emerging Networking Experiments and Technologies, 2009, pp. 1–12.
[7] J. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Communications Magazine 49 (7) (2011) 26–36, doi:10.1109/MCOM.2011.5936152.
[8] W. Li, S. Oteafy, H. Hassanein, Rate-selective caching for adaptive streaming over information-centric networks, IEEE Transactions on Computers PP (99) (2017) 1, doi:10.1109/TC.2017.2687920.
[9] T. Stockhammer, Dynamic adaptive streaming over http –: Standards and design principles, in: Proceedings of the Second Annual ACM Conference on Multimedia Systems, in: MMSys '11, ACM, New York, NY, USA, 2011, pp. 133–144, doi:10.1145/1943552.1943572.
[10] C. Müller, S. Lederer, C. Timmerer, An evaluation of dynamic adaptive streaming over http in vehicular environments, in: Proceedings of the 4th Workshop on Mobile Video, in: MoVid '12, ACM, New York, NY, USA, 2012, pp. 37–42, doi:10.1145/2151677.2151686.
[11] S. Zhao, Z. Li, D. Medhi, P. Lai, S. Liu, Study of user qoe improvement for dynamic adaptive streaming over http (mpeg-dash), in: 2017 International Conference on Computing, Networking and Communications (ICNC), 2017, pp. 566–570, doi:10.1109/ICCNC.2017.7876191.
[12] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, H. Hellwagner, An experimental analysis of dynamic adaptive streaming over http in content centric networks, in: 2013 IEEE International Conference on Multimedia and Expo (ICME), 2013, pp. 1–6, doi:10.1109/ICME.2013.6607500.
[13] B. Rainer, D. Posch, H. Hellwagner, Investigating the performance of pull-based dynamic adaptive streaming in ndn, IEEE Journal on Selected Areas in Communications 34 (8) (2016) 2130–2140, doi:10.1109/JSAC.2016.2577365.
[14] J. Samain, G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara, M. Tortelli, D. Rossi, Dynamic adaptive video streaming: Towards a systematic comparison of icn and tcp/ip, IEEE Transactions on Multimedia 19 (10) (2017) 2166–2181, doi:10.1109/TMM.2017.2733340.
[15] M.F. Majeed, S.H. Ahmed, S. Muhammad, H. Song, D.B. Rawat, Multimedia streaming in information-centric networking: A survey and future perspectives, Computer Networks 125 (Supplement C) (2017) 103–121, doi:10.1016/j.comnet.2017.05.030. Softwarization and Caching in NGN
[16] S. Lederer, C. Mueller, C. Timmerer, H. Hellwagner, Adaptive multimedia streaming in information-centric networks, IEEE Network 28 (6) (2014) 91–96, doi:10.1109/MNET.2014.6963810.
[17] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, H. Hellwagner, Adaptive streaming over content centric networks in mobile networks using multiple links, in: 2013 IEEE International Conference on Communications Workshops (ICC), 2013, pp. 677–681, doi:10.1109/ICCW.2013.6649319.
[18] S. Petrangeli, N. Bouten, M. Claeys, F.D. Turck, Towards svc-based adaptive streaming in information centric networks, in: 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2015, pp. 1–6, doi:10.1109/ICMEW.2015.7169859.
[19] Y. Liu, J. Geurts, J.C. Point, S. Lederer, B. Rainer, C. Mueller, C. Timmerer, H. Hellwagner, Dynamic adaptive streaming over ccn: A caching and overhead analysis, in: 2013 IEEE International Conference on Communications (ICC), 2013, pp. 3629–3633, doi:10.1109/ICC.2013.6655116.
[20] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, A. Bragagnini, Offloading cellular networks with information-centric networking: The case of video streaming, in: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012, pp. 1–3, doi:10.1109/WoWMoM.2012.6263734.
[21] M. Conti, R. Droms, M. Hassan, S. Valle, Qoe degradation attack in dynamic adaptive streaming over icn, IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2018, (In press).
[22] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Transactions on Information Theory 52 (10) (2006) 4413–4430, doi:10.1109/TIT.2006.881746.
[23] M.-J. Montpetit, C. Westphal, D. Trossen, Network coding meets information-centric networking: An architectural case for information dispersion through native network coding, in: Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications, in: NoM '12, ACM, New York, NY, USA, 2012, pp. 31–36, doi:10.1145/2248361.2248370.
[24] W.-X. Liu, S.-Z. Yu, G. Tan, J. Cai, Information-centric networking with built-in network coding to achieve multisource transmission at network-layer, Computer Networks 115 (2017) 110–128.
[25] C. Kreuzberger, D. Posch, H. Hellwagner, Amust framework - adaptive multimedia streaming simulation framework for ns-3 and ndnsim, 2016.

[26] S. Lederer, C. Müller, C. Timmerer, Dynamic adaptive streaming over http dataset, in: Proceedings of the 3rd Multimedia Systems Conference, in: MMSys '12, ACM, New York, NY, USA, 2012, pp. 89–94, doi:10.1145/2155555.2155570.
[27] Y. Liu, J.Y.B. Lee, A unified framework for automatic quality-of-experience optimization in mobile video streaming, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9.
[28] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, M. Luby, User experience modeling for dash video, in: 2013 20th International Packet Video Workshop, 2013, pp. 1–8.
[29] C. Kreuzberger, D. Posch, H. Hellwagner, A scalable video coding dataset and toolchain for dynamic adaptive streaming over http, in: T.O. Wei (Ed.), Proceedings of the 6th ACM Multimedia Systems Conference, ACM, New York, NY, USA, 2015, pp. 213–218.
[30] S. Lederer, C. Mueller, C. Timmerer, Dynamic adaptive streaming over HTTP dataset, in: M. Claypool, C. Griwodz, K. Mayer-Patel (Eds.), Proceedings of the Third Annual ACM SIGMM Conference on Multimedia Systems (MMSys12), ACM, New York, NY, USA, 2012, pp. 89–94.
[31] J. Jiang, V. Sekar, H. Zhang, Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive, IEEE/ACM Transactions on Networking 22 (1) (2014) 326–340, doi:10.1109/TNET.2013.2291681.
[32] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A.C. Begen, D. Oran, Probe and adapt: Rate adaptation for http video streaming at scale, IEEE Journal on Selected Areas in Communications 32 (4) (2014) 719–733, doi:10.1109/JSAC.2014.140405.
[33] C. Sieber, T. Hofeld, T. Zinner, P. Tran-Gia, C. Timmerer, Implementation and user–centric comparison of a novel adaptation logic for dash with svc, in: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), 2013, pp. 1318–1323.
[34] S. Akhshabi, S. Narayanaswamy, A.C. Begen, C. Dovrolis, An experimental evaluation of rate-adaptive video players over http, Image Commun. 27 (4) (2012) 271–287.
[35] A. Compagno, M. Conti, P. Gasti, L.V. Mancini, G. Tsudik, Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking, Springer International Publishing, Cham, pp. 243–262.
[36] G. Acs, M. Conti, P. Gasti, C. Ghali, G. Tsudik, C. Wood, Privacy-aware caching in information-centric networking, IEEE Transactions on Dependable and Secure Computing PP (99) (2017).
[37] F. Zhang, W. He, X. Liu, P.G. Bridges, Inferring users' online activities through traffic analysis, in: Proceedings of the Fourth ACM Conference on Wireless Network Security, in: WiSec '11, ACM, New York, NY, USA, 2011, pp. 59–70, doi:10.1145/1998412.1998425.
[38] M. Liberatore, B.N. Levine, Inferring the source of encrypted http connections, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, in: CCS '06, ACM, New York, NY, USA, 2006, pp. 255–263, doi:10.1145/1180405.1180437.
[39] J. Saltarin, E. Bourtsoulatze, N. Thomos, T. Braun, Adaptive video streaming with network coding enabled named data networking, IEEE Transactions on Multimedia 19 (10) (2017) 2182–2196, doi:10.1109/TMM.2017.2737950.
[40] M. Bilal, S. Kang, Network-coding approach for information-centric networking, IEEE Systems Journal 13 (2) (2019) 1376–1385.
[41] T. Ho, M. Mdard, J. Shi, M. Effros, D.R. Karger, On randomized network coding, in: In Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing, 2003.
[42] J. Saltarin, E. Bourtsoulatze, N. Thomos, T. Braun, Netcodccn: A network coding approach for content-centric networks, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9, doi:10.1109/INFOCOM.2016.7524382.
[43] A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM: NDN simulator for NS-3, Technical Report, NDN, 2012.
[44] C. Mueller, S. Lederer, J. Poecher, C. Timmerer, Demo paper: Libdash - an open source software library for the mpeg-dash standard, in: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2013.
[45] M.V. Pedersen, J. Heide, F.H.P. Fitzek, Kodo: An open and research oriented network coding library, in: V. Casares-Giner, P. Manzoni, A. Pont (Eds.), NETWORKING 2011 Workshops, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 145–152.
[46] N. Spring, et al., Measuring ISP Topologies with Rocketfuel, IEEE/ACM Trans. Netw. (2004).
[47] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, P. Halvorsen, Flicker effects in adaptive video streaming to handheld devices, in: Proceedings of the 19th ACM International Conference on Multimedia, in: MM '11, ACM, New York, NY, USA, 2011, pp. 463–472, doi:10.1145/2072298.2072359.

Muhammad Hassan completed the Bachelors in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Pakistan in 2008 and the Masters in Computer Network Engineering from HALMSTAD University, Sweden in 2013. Currently, he is a PhD student in Brain, Mind and Computer Science at the University of Padua, Italy. He is also a part of the SPRITZ Security and Privacy research group, Padua. His research interests are in the area of securing ICN architecture and related studies such as secure integration of existing technologies in future Internet architecture.

Mauro Conti is Full Professor at the University of Padua, Italy, and Affiliate Professor at the University of Washington, Seattle, USA. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Postdoc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU (2008, 2016), UCLA (2010), UCI (2012, 2013, 2014, 2017), TU Darmstadt (2013), UF (2015), and FIU (2015, 2016, 2018). He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of security and privacy. In this area, he published more than 250 papers in topmost international peer-reviewed journals and conference. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, and General Chair for SecureComm 2012 and ACM SACMAT 2013. He is Senior Member of the IEEE.

Chhagan Lal is currently working as a postdoctoral research fellow at Simula Research Laboratory, Norway. Previously, he was a postdoctoral fellow at Department of Mathematics, University of Padua, Italy, where he was part of the SPRITZ research group. He received his PhD in Computer Science and Engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. During his PhD, he has been awarded with the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in University of Saskatchewan, Saskatoon, SK, Canada. His current research areas include applications of blockchain technologies, security in softwaredefined networking, and Internet of Things networks.